一种自适应无线局域网协议*

彭 泳+, 程时端

(北京邮电大学 程控交换与通信网国家重点实验室,北京 100876)

A Self-Adaptive Wireless LAN Protocol

PENG Yong+, CHENG Shi-Duan

(National Laboratory of Switching Technology and Telecommunication Networks, Beijing University of Posts and Telecommunications, Beijing 100876, China)

+ Corresponding author: Phn: +86-10-62282007, E-mail: py_hello@yahoo.com.cn, http://www.butp.edu.cn

Received 2003-06-03; Accepted 2003-07-23

Peng Y, Cheng SD. A self-adaptive wireless LAN protocol. *Journal of Software*, 2004,15(4):604~615.
http://www.jos.org.cn/1000-9825/15/604.htm

Abstract: This paper presents an in-depth analysis of the DCF (distributed coordination function) access mode of IEEE802.11 protocol. Based on the result of this study, the authors have developed a new self-adaptive wireless LAN MAC (medium access control) algorithm with the name of NSAD (new self-adaptive DCF algorithm). Simulation results show that NSAD is superior to DCF in all characters concerned (e.g. goodput, fairness, and packet drop rate).

Key words: DCF (distributed coordination function); wireless LAN; MAC (medium access control)

摘 要: 研究了无线局域网 IEEE802.11 协议的 DCF(distributed coordination function)接入方式,在建模基础上进行算法改进,提出了一种新的节点自适应链路碰撞的退避算法 NSAD(new self-adaptive DCF algorithm).大量的仿真实验表明,所提出的算法在吞吐量、公平性、丢包等方面较原 DCF 协议都有显著提高.

关键词: DCF(distributed coordination function);无线局域网;MAC

中图法分类号: TP393 **文献标识码**: A

近年来,由于移动通信给人们带来方便、快捷的服务,有关研究也越来越受到人们的重视.无线局域网(wireless local area network)作为一种重要的移动主机实现本地接入的方法,逐渐成为研究的热门课题.为此,IEEE 制定了无线局域网的标准 IEEE802.11 以提供宽带的、支持异步或同步服务的网络[1].DCF(distributed coordination function)是 IEEE802.11 协议的基本接入方式,它通过窗口指数退避来实现不同站点的异步自适应接入.这种接入方式协议实现比较简单,因此得到目前大多数厂商的支持,但研究表明,在节点较多的情况下,链

* Supported by the National Natural Science Foundation of China under Grant No.90204003 (国家自然科学基金); the National Research Foundation for the Doctoral Program of Higher Education of China under Grant No.20010013003 (国家教育部博士点基金); the China Postdoctoral Science Fo-Undation under Grant No.2003034111 (中国博士后科学基金)

作者简介: **彭泳**(1978-),男,湖北赤壁人,博士生,主要研究领域为无线网络,网络服务质量;**程时端**(1940-),女,教授,博士生导师,主要研究领域为通信网与计算机网的体系结构和性能分析,互联网的服务质量控制和管理.

路吞吐量、公平性均恶化明显[2].以前的一些研究者为了提高 802.11DCF 吞吐量和公平性提出了一些改进措施(例如文献[3~10]),主要可以分为以下两类:一类是通过建模分析推导来获取网络最大吞吐量,如文献[4,5,8,9];另一类主要是通过改变窗口调节机制和各种 DCF 参数来控制节点的接入以达到网络性能的优化,如文献[3,6~8].

本文首先引入一个与链路负荷状态相关的参数 l,通过分析推导发现,通过调节节点接入概率使得链路吞吐量达到最大时,l 的最优值 l_{opt} 具有与当前活动节点数量近似无关的特性.我们利用这个特性改进 DCF 并提出了一种新型自适应算法 NSAD(new self-adaptive DCF algorithm).大量仿真实验结果表明,NSAD 协议的窗口调节机制能够自适应当前链路状况,在节点较多时依然保持较高的吞吐量和较好的公平性.

本文第 1 节介绍算法相关的技术背景.第 2 节描述了 NSAD 的具体实现流程,并利用数学建模分析推导了 l_{opt} 在节点数较多时具有与当前活动节点数量近似无关的特性.第 3 节和第 4 节描述了仿真环境和仿真模型的参数,并对仿真结果作了分析和说明.最后在第 5 节中给出结论.

1 相关技术背景

1.1 IEEE802.11 DCF接入方式

IEEE 在制定无线局域网标准时,针对不同的应用,在 802.11 的 MAC(medium access control)层协议给出了两种接入模式:异步模式 DCF 和同步模式 PCF(point coordination function)[1].DCF 是实现 PCF 的基础.DCF 模式下主机以 CSMA(carrier sense multiple access)/CA(collision avoidance)的方式独立接入信道,通过指数退避算法减少碰撞概率.主机对信道的检测、发送分组以及指数退避算法都以时隙(slot)为基本单位,所以 DCF 本质上是一个时隙的 MAC 协议.

为了减小碰撞的损失以及隐藏终端等的影响,在 IEEE802.11 的 DCF 算法中可以选择使用 RTS(request to send)和 CTS(clear to send)来预留信道带宽.RTS 与 CTS 长度(RTS 为 20 字节,CTS 为 14 字节)相比最大数据帧长(2 346 字节)较短,因此发生碰撞花费的时间会相对较少.在 RTS 和 CTS 帧里存放预留时长信息.接收到 RTS/CTS 的基站获取这个信息后更新自己的 NAV(network allocation vector)值,为竞争信道的基站进行数据交换预留出信道带宽.图 1 显示了 IEEE802.11 协议利用 RTS/CTS 传送数据的流程图.

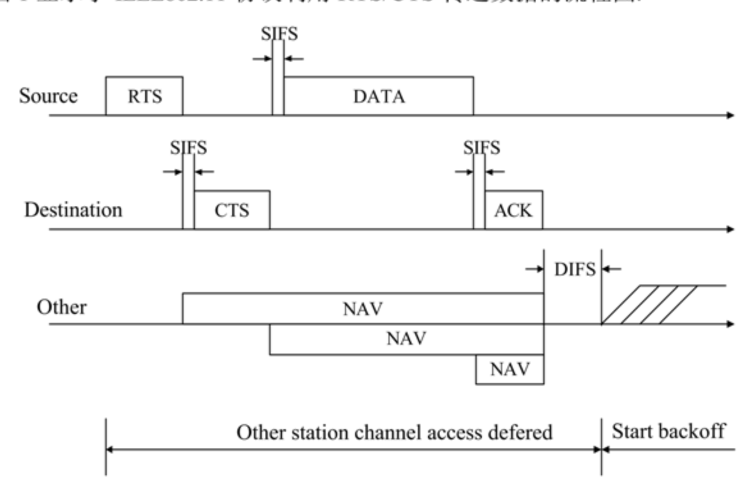


Fig.1 Use RTS/CTS to reserve channel and exchange data frame

图 1 利用 RTS/CTS 预留信道传送数据帧示意图

从图 1 中可以看出,如果主机有数据包要发送,首先会等待信道空闲 DIFS(DCF InterFrame space),之后启动 BACKOFF 退避定时器,在信道空闲时每隔一个时隙计数器递减 1.当计数器减为 0 时发送 RTS 竞争信道.当检测到信道忙的时候,冻结退避定时器.如果多个主机的 RTS 发生交叠,则被认为是发生碰撞.此时,发送 RTS 的主机不会收到 CTS,超时定时器经过 SIFS(short InterFrame space)+CTS+DIFS 之后会判断竞争失败并进行窗口退

避,进行下一轮竞争.如果竞争信道成功,主机就转入发送数据帧.对于数据发送成功的判断是:只有收到数据帧的确认后,才能认为相应的数据帧发送成功,之后重置窗口,并从上层获取数据包,准备进入下一轮发送;假如没有收到确认,则认为发送失败,之后主机进行指数退避,选择一个退避时间,等待信道空闲并再次竞争信道.传统DCF 窗口退避规则如下[1](W 表示 MAC 层竞争窗口,W_{min},W_{max}表示窗口值的系统设定上下界):

a) 初始窗口值的计算

$$W=W_{min} \tag{1}$$

b) 窗口指数退避

$$W=W\times2+1;\quad \text{if } (W>W_{max})\ W=W_{max} \tag{2}$$

c) 退避时长计算

$$\text{Backoff Time}=\text{Random}()\times a\text{SlotTime},\ \text{Random}()\text{均匀分布于}[0,W] \tag{3}$$

一些研究者为了提高 802.11DCF 吞吐量和公平性提出了一些改进,主要可以分为以下两类:一类是通过建模分析推导来获取网络最大吞吐量,如文献[4,5,8,9].文献[4]在简单建模的基础上通过分析退避窗口、当前主机数等参数之间的关系力图达到链路吞吐量的最优化.文献[5]通过网络碰撞时段和网络空闲时段平衡的思想推导网络最优化吞吐量.文献[8,9]对IEEE802.11 的 DCF 方式利用马尔科夫模型进行建模,文献[9]在建模的基础上推导了饱和吞吐量,文献[8]对文献[9]的建模进行了一些修正.然而他们在对网络改进时大多是通过预测网络中的活动节点数来调节网络使其达到最优.但是由于无线网络随机接入的不确定性以及无线链路丢包频繁,进行节点数量预测会引入不必要的误差.

另一类主要是通过改变窗口调节机制和各种 DCF 参数来控制节点的接入以达到网络性能的优化,如文献[3,6~8].文献[3]分析了无线局域网中存在的公平性的问题,并提出了一些改进公平性的算法,例如为了避免少数节点窗口过大而导致的不公平性而使用的交换窗口机制等.文献[6~8]通过针对不同应用赋予不同 DCF 参数以期达到提高吞吐量、保证 QoS 的目的.

1.2 CSMA/CA协议吞吐量最大化准则

设当前所有主机在一个单区网中,各主机通过 DCF 方式接入自发组成 Ad Hoc 网络.一些研究者曾经涉及到在这种环境下的吞吐量最大化的准则,并通过仿真分析得出了一些结论.文献[4,9]用马尔科夫模型建模,推导出了站点数和近似最优窗口值的公式.文献[5,11]采用以下思路推导最优化吞吐量:如图 2 所示,*t_v* 是成功传送一个数据包所经历的时长.*t_coll* 代表在 *t_v* 时间段内碰撞所花费的时间,*t_free* 代表 *t_v* 时间段内由于退避而产生的空闲时间段长度,*t_pack* 代表成功发送数据帧的开销,*t_succ* 代表其他站点成功发送的开销.显然,*t_coll* 和 *t_free* 是 CSMA/CA 共享信道竞争所带来的开销.在这样的定义下,网络利用率不高表现为以下两种情况:

情况 1. 信道过于空闲.此时 *t_free* 相对于 *t_coll* 过多.出现这种情况的原因是当前拥塞窗口相对于当前信道的主机数过大,造成不必要的退避时延.

情况 2. 信道过于拥塞.此时 *t_coll* 相对于 *t_free* 过多.出现这种情况的原因是当前拥塞窗口相对于当前信道的主机数过小,在一个窗口时间内过多的主机竞争信道,造成碰撞过于频繁.在这种情况下,发生退避的主机竞争窗口在短期内往往维持在一个较高值,主机成功接入的概率偏小;成功发送的主机窗口由于经常重置刷新,总保持一个相对较低的值,因此总能以较大概率接入信道,最终导致链路公平性急剧下降.

为了避免以上情况的发生,应该使 *t_idle* 和 *t_free* 的比值趋向最优值.当我们调节站点接入信道的概率(在本文中为初始窗口大小)使得比值趋向最优值时,协议能够获得最大吞吐量.我们下面的设计就是依据以上原则进行的.

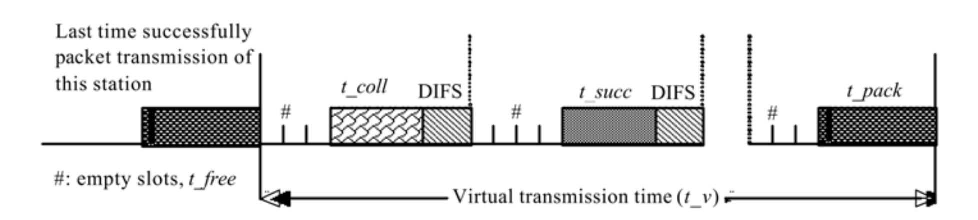


Fig.2 Procedure of one successful data packet transmission
图 2 一次数据包成功发送所经历的过程

2 改进的 DCF 协议: NSAD (new self-adaptive DCF algorithm)

2.1 NSAD协议描述

由第 1 节中的分析可知,传统的 DCF 协议流程在传送包时并没有估计当前链路的拥塞状况.在网络节点数较多时,节点依然使用以前的初始竞争窗口接入信道,造成碰撞过多和频繁退避,最终导致链路的吞吐量和公平性恶化明显[2].

NSAD 为了让协议能够自适应当前链路状况,引入了一个与链路碰撞密切相关的参数 l,以反映当前链路负载状况.每个移动站点独立地根据当前网络碰撞时长和空闲时长计算当前 l 参数的大小.根据获得的负载状况 l,与最优的负载值 l_{opt} 相比较,并通过比较结果来进一步调节站点初始接入窗口的大小.算法改动包括 l 值的计算和新的窗口退避规则.

定义. 链路负载 l 定义为:一次成功传送数据所经历的碰撞平均时长与空闲平均时长之比.

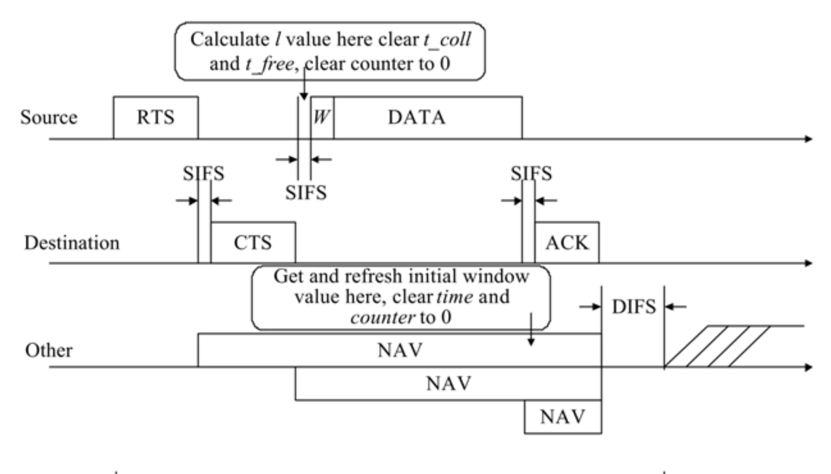
每次成功的数据传输会触发计算一次成功传送数据所经历的平均碰撞时长和空闲时长,分别用 *t_coll_avg* 和 *t_free_avg* 表示.在本次数据传送过程(即图 2 中的 virtual transmission time)中经历的碰撞时长和空闲时长分别表示为 *t_coll* 和 *t_free*.l 值的计算方法如下:

$$t_coll_avg=\lambda\times t_coll_avg+(1-\lambda)\times t_coll \tag{4}$$

$$t_free_avg=\lambda\times t_free_avg+(1-\lambda)\times t_free \tag{5}$$

$$l=t_coll_avg/t_free_avg \tag{6}$$

l 初始值设为优化值 l_{opt}(通过建模计算 l_{opt},在第 2.2 节中会有详细说明).λ为平滑随机抖动参数,在本文中取值 0.9~0.95 之间.如图 3 所示,每当主机利用 RTS/CTS 竞争获得信道时,根据收集信道状况所得的 *t_coll* 和 *t_free* 分别计算其平均值和当前网络负载 l 值,并与最优值 l_{opt} 比较,判断调节竞争窗口大小.我们设定门限值σ以触发判断是否当前网络的 l 值在期望范围以内,并设定游动计数器 *counter*.如果($l>l_{opt}+\sigma$),则触发 *counter* 增加 1;如果($l<l_{opt}-\sigma$),则触发 *counter* 减小 1.如果在成功传送 M 次,发现统计获得的 *counter* 绝对值大于一个预设值 MAX_*counter* 时触发窗口调节机制,并将调节后的新的优化窗口值通过数据包捎带到当前网络所有其他站点.考虑到无线网络中存在较高的误码率,同时随机接入的站点也不可能马上获得优化窗口值,因此网络中必然存在初始窗口值不同的站点.由于小窗口值的站点具有较高的接入概率,同时虚拟传送时长也较短,如果设定固定 M 值,必然会使小初始窗口值的站点与大初始窗口值窗口更新周期 T_p 的不公平,导致当前网络窗口值相对于最优窗口值总是偏小.因此需要根据不同初始窗口值 W_{init},通过碰撞和退避估算相应的 M 值大小,保证当前网络中所有活动站点的 T_p 近似相当.M 值的计算详见附录.



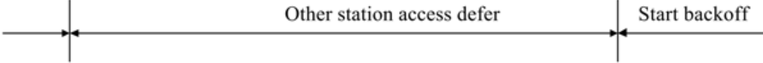


Fig.3 The calculation of load factor and initial contention window refresh

图 3 NSAD 计算负载值并刷新初始竞争窗口

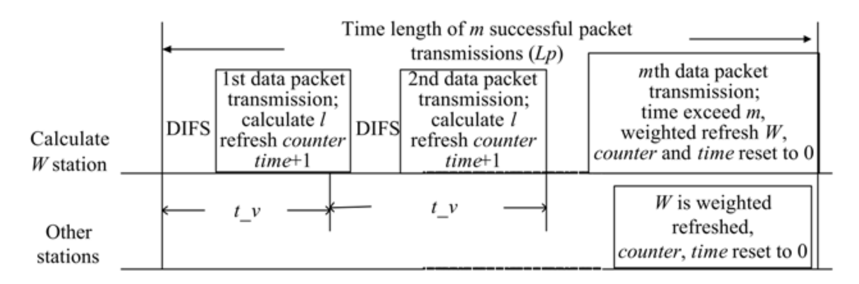


Fig.4 Initial contention window refresh mechanism

图 4 初始窗口刷新机制

窗口调节机制如图 4 及下面算法所示.初始竞争窗口调节范围设定在[W_{min},(W_{max}−1)/2].

NSAD 新初始窗口调节算法.

```
if (l>l_opt+σ) counter=counter+1;                //σ is the threshold that stimulate counter change
if (l<l_opt−σ) counter=counter−1;                //counter count forward or backward according to l
if (time<m) time=time+1                          //count transmission time
else {
    time=0;
    //Broadcast the new initial window W_init;
    if (counter>MAX_counter){                    //contention window is small
        //relative to node number
        counter=0;                               //refresh counter
        W_init=(W_init+1)×2−1;                   //multiple increase initial window
    }else if (counter<−MAX_counter){             //contention window is large
        //relative to node number
        counter=0;                               //refresh counter
        W_init=(W_init+1)/2−1;                   //multiple decrease initial window
    }else
```

```
    {Winit=Winit;};
    if (Winit>(Wmax+1)/2−1)      Winit=(Wmax+1)/2−1;      //up bound of initial window
    if (Winit<Wmin)              Winit=Wmin;              //low bound of initial window
};
```

2.2 l_{opt}的确定

首先设定当前单跳网络活动节点数为 N.所有站点以 W 的初始窗口大小接入无线信道.站点在每个空闲时槽试图接入信道的概率为τ.显然,τ随 W 的不同而有所改变.最大化吞吐量时最优窗口值为 W_{opt},对应的接入概率为τ_{opt}.我们首先来推导 l_{opt} 应该满足的条件.T_{coll},T_c^*分别代表碰撞的平均时长和碰撞平均时槽数,T_{slot} 代表 一 个时槽的时长,T_{data},T_d^*分别为数据帧的时长和时槽数,T_{busy},T_b^*分别为成功竞争到信道忙的总时长和时槽数.P_{coll}为碰撞发生的概率,P_{free} 是时槽空闲概率.P_{succ} 是某移动站点在一个空闲时槽成功竞争到信道的概率,亦即只有一个站点在此空闲时槽竞争信道的概率.依概率显然有

$$\left.\begin{aligned} P_{coll} &= 1-N\tau(1-\tau)^{N-1}-(1-\tau)^N \\ P_{free} &= (1-\tau)^N \\ P_{succ} &= N\tau(1-\tau)^{N-1} \end{aligned}\right\} \tag{7}$$

$$l=\frac{T_{coll}\cdot P_{coll}}{T_{slot}\cdot P_{free}}=\frac{1-N\tau(1-\tau)^{N-1}-(1-\tau)^N}{(1-\tau)^N}\cdot\frac{T_c^*}{1} \tag{8}$$

当吞吐量 S 达到最大值的时候,也就是下式达到最大值的时候:

$$S=\frac{P_{succ}\cdot T_{data}}{P_{succ}\cdot T_{busy}+P_{free}\cdot T_{slot}+P_{coll}\cdot T_{coll}} \tag{9}$$

由文献[11]可得,要使 S 最大,需满足下面的方程:

$$(1-\tau)^N+T_c^*\cdot\left[1-N\tau-(1-\tau)^N\right]=0 \tag{9}$$

注意到 $\tau\to 0$, $(1-\tau)^N\approx 1-N\tau+\frac{N(N-1)}{2}\tau^2$,带入上式得:

$$\tau_{opt}=\frac{\sqrt{[N+2(N-1)(T_c^*-1)]/N}-1}{(N-1)(T_c^*-1)} \tag{10}$$

将文献(10)中τ_{opt}代入式(8),获得下式:

$$l_{opt}=\left[\frac{1-N\tau_{opt}(1-\tau_{opt})^{N-1}-(1-\tau_{opt})^N}{(1-\tau_{opt})^N}\right]\cdot T_c^* \tag{11}$$

式(10)和式(11)表明了当前无线信道中活动站点数与最优负载值的关系式.图 5 和图 6 分别显示了 l_{opt} 和当前活动节点数的关系曲线以及 l_{opt} 和平均碰撞时槽数关系曲线.

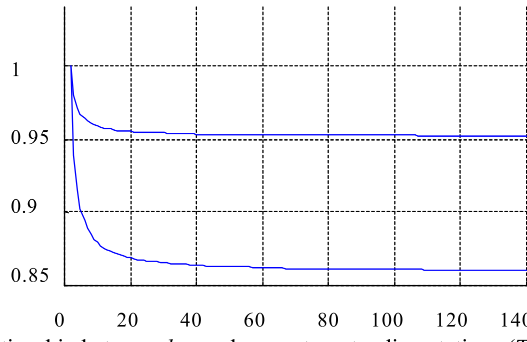


Fig.5 The relationship between l_{opt} and current contending stations (T_c^*=29.0, T_c^*=331.8)

图 5 l_{opt} 和当前活动节点数的关系曲线(T_c^*=29.0 和 T_c^*=331.8)

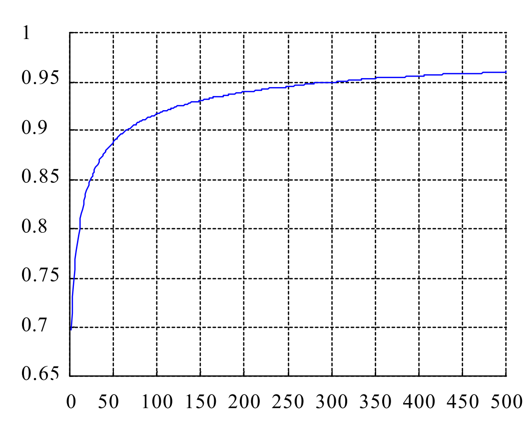


Fig.6 The relationship between l_{opt} and average collision slots (140 active nodes)

图 6 l_{opt}和平均碰撞时槽数关系曲线(活动节点数为 140)

根据图 5 我们发现,当节点数大于 20 的时候,l_{opt}几乎保持一个衡定值,这个值与 T_c^*相关,而与节点数无关.图 5 和图 6 的重要性在于,为当前共享竞争信道的所有站点提供了一个衡量当前网络是否过载的尺度.与以往一些算法不同,我们的算法不需要预测当前网络中的具体活动节点个数,直接感知当前网络中忙的时长和空闲的时长,从而避免了预测节点所带来的不准确性.当使用 RTS/CTS 选项时,碰撞时长可以计算为 RTS+EIFS 的时长.当采用 DSSS 2Mbps 的物理链路环境传输时,T_c^*=29.0(slots),根据图 5,l_{opt}=0.86.当 1 500 字节的 TCP 数据帧发生碰撞时,碰撞时长可计算为 DATA+SIFS+ACK+DIFS,于是 T_c^*=331.8(slots).根据图 5,l_{opt}=0.95.尽管上述讨论中是以使用 RTS/CTS 选项为例进行的,但是我们的机制也适用于不使用 RTS/CTS 选项的环境.由于在当前单跳网络中,只有成功传送的数据包可以将新的优化窗口值捎带给所有的相邻节点,所以无论数据包是否成功传送,我们都可以将优化窗口值写入数据包.如果数据包成功传送,发送方会收到 ACK 并知道新的优化窗口值被带给所有其他相邻站点,此时发送方可重置本地计数器准备下一轮网络状态测量;如果数据包发送失败,则发送方在等待 ACK 超时后得知发送失败,将碰撞加入碰撞时长计数并继续本轮碰撞时长统计.

由于 l_{opt}是一个与活动站点数无关,只需根据碰撞时长预测的值,我们利用这一点来调节初始竞争窗口 W 的大小,使得 l 接近最优值 l_{opt}.调节的结果使得 W 收敛到一个对应于最优接入概率 τ_{opt} 的最优值 W_{opt},同时,网络吞吐量达到最大.

2.3 最优窗口值W_{opt}与活动节点数N的关系

初始竞争窗口值为 W_{init},窗口上界和下界分别为 W_{max}和 W_{min}.令 $n=\log_2[(W_{max}+1)/(W_{init}+1)]$,实际上 n 是在发生碰撞退避时竞争窗口增大的次数.n'是系统预设的最大退避次数(IEEE802.11 中短帧重传次数为 7).p 是站点竞争信道遭遇碰撞的条件概率.关于 W_{opt} 和 N,我们有以下结论:

定理 1. 当前活动站点数 N 和优化初始竞争窗口值 W_{init} 之间的关系式可用下列方程表示.详细证明见附录.

$$N\cdot\sqrt{2T_c^*}=\frac{\sum_{i=0}^{n'}p^i+(W_{init}+1)\sum_{i=0}^{n-1}(2p)^i+(W_{max}+1)\sum_{i=n}^{n'}p^i}{\sum_{i=0}^{n'}p^i} \tag{12}$$

站点接入信道碰撞概率就等于存在其他站点在同一时槽试图接入信道的概率.显然依概率有

$$p=1-(1-\tau)^{N-1} \tag{13}$$

由式(10)可得[9]: $\tau=\tau_{opt}\approx\dfrac{1}{N\sqrt{T_c^*/2}}$,在式(13)中我们用τ_{opt}来替代τ,当 N 足够大时,使用近似: $\lim\limits_{x\to\infty}\left(1+\dfrac{1}{x}\right)^x=\mathrm{e}$,于是得到:

$$p \approx 1 - e^{-\frac{1}{\sqrt{T_c^*/2}}} \quad (14)$$

上式表明,当 N 足够大时,碰撞概率 p 近似为一个恒定值.

通过方程组(12)和方程组(14),我们可以获得最优初始窗口值 W_{init} 与当前无线网络活动节点个数 N 的关系式.令 n=1,2, …,5,W_{max}=1 023,T_c^*=29.0(for RTS/CTS collision),n'=7,代入式(12),我们可以得到 N 与 W_{init} 之间的变化关系表.非常有意思的是(见表 1),数值计算表明优化后的初始窗口值 W_{init} 通过计算与当前网络活动节点数 N 具有近似的线性关系.以上在 NSAD 协议中通过最优初始窗口值的大小计算预测当前无线网络活动站点数,从而获取当前网络负载状况,对于合理利用网络资源以及设计合理的负载均衡算法提供了便利.

Table 1 Relationship between optimal W_{init} and the number of contending stations

表 1 最优初始窗口值 W_{init} 与当前无线网络活动节点个数的关系表

W_{init}	n	N
31	5	6
63	4	12
127	3	23
255	2	45
511	1	83

3 仿真模型

为验证第 2 节中所述算法的有效性,我们采用了 Berkeley 大学的 NS(network simulator)仿真工具进行网络仿真,并拟定以下仿真环境.为了简化模型,我们首先作以下假设[10]:

(1) 因为我们研究的重点是多点接入中的控制问题,所以假设所有站点可达,不考虑隐藏终端(hidden terminal)和暴露终端(exposed terminal)的问题.

(2) 虽然发生冲突时接收机可以捕获信号最强的帧,但在这里,我们为了讨论方便,采取保守认为只要发生碰撞就意味着发送失败的思想.

(3) 缓存区足够大,帧丢失只是因为碰撞和超时引起的,不会由缓冲区不足引起.对于主机而言容易做到.

仿真拓扑采用的是设置 N 个随机分布的节点.在第奇数号和第偶数号节点之间建立 TCP 连接,开设 FTP 服务,如图 7 所示.在仿真过程中使奇数号节点和偶数号节点均匀分布于通信范围以内.TCP 版本使用的是 New Reno.网络拓扑和主要参数设置见表 2.

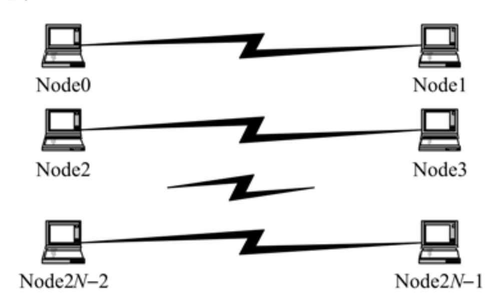


Fig7 Simulation topology of N nodes

图 7 N 个节点仿真拓扑

Table 2 Simulation model parameter set

表 2 仿真模型参数设置

Channel bit rate (bps)	2M
Number of nodes	4,10,30,50,70,100,140
Window upper and lower bound	31,1 023
FTP start and stop time (s)	10.0,135.0
l_{opt}	0.85 (RTS/CTS)
Window adjustment threshold σ	0.3
MAX_*counter*	$\lfloor M/2 \rfloor+1$
Data payload length (bytes)	1500

4 仿真结果及分析

4.1 吞吐量和公平性分析

图 8 是系统吞吐量和节点数之间关系曲线图.图中每个点都是经过 10 次取不同种子仿真结果的均值.从图中可以看出,随着节点数的增加,NSAD 吞吐量较之原 DCF 有较大的提高.节点数越多,性能提高越明显.节点数为 140 的时候,吞吐量提高达到 40%.节点数较多时,NSAD 方差较小,表明稳定性有所提高.

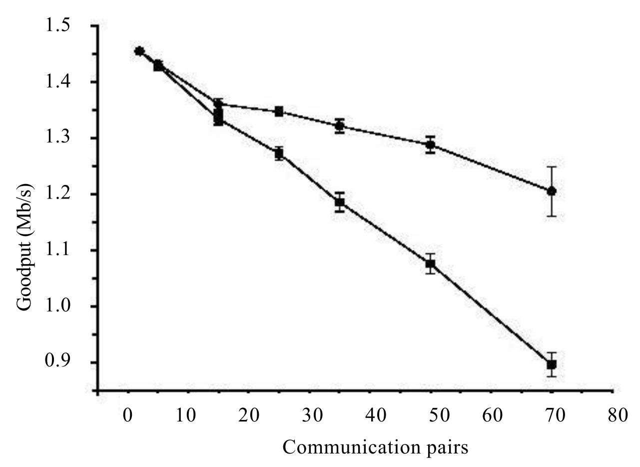


Fig.8 Relation between goodput and number of contending stations

图 8 吞吐量和节点数之间关系曲线图

我们也提取了一幅关于协议公平性的曲线图.我们使用公式(15)作为公平性的计算公式.$f(i)$表示第 i 个通信对之间的吞吐量.在图 9 中我们可以看到,协议公平性在节点数增多时提高较为明显.

$$f=\left(\sum_{i=1}^{n} f(i)\right)^2 \bigg/ \left(n\sum_{i=1}^{n} f(i)^2\right) \tag{15}$$

4.2 丢包分析

从图 9 和图 10 可以看到,DCF 在重负载下丢包随时间呈线性增长,而 NSAD 只是在刚开始,窗口未能调节到最优的时候发生丢包,之后就没有丢包了.由此可知,NSAD 的窗口调节机制能够很好地屏蔽底层链路由于共享信道站点之间的竞争造成的重传丢包.

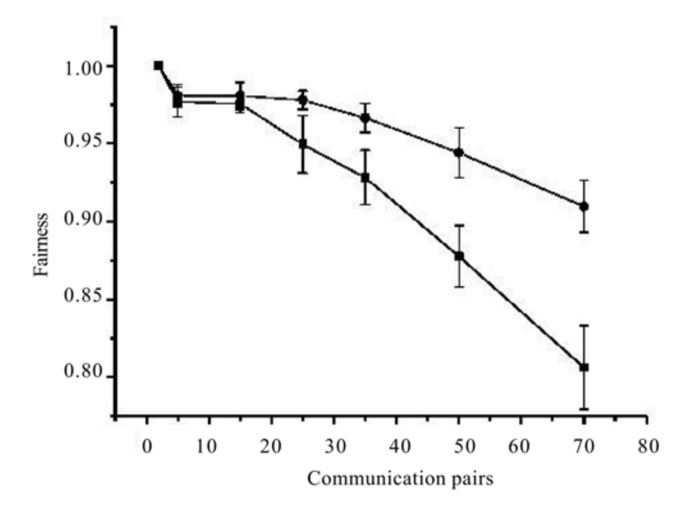


Fig.9 Relationship between fairness and the number of contending stations

图 9 公平性和节点数之间关系曲线图

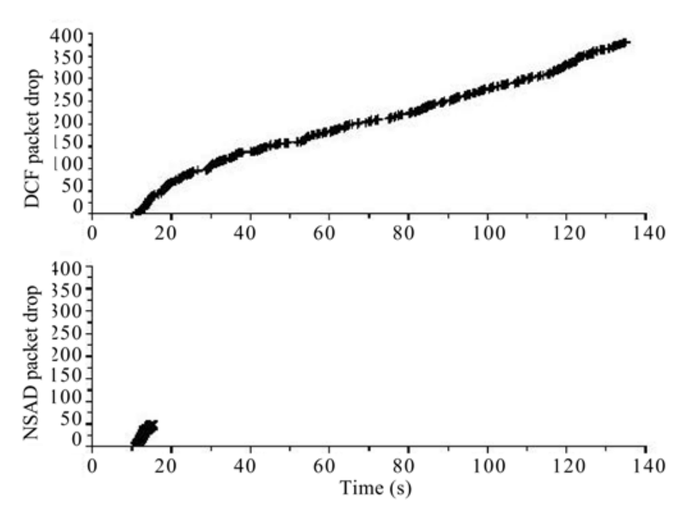


Fig.10 DCF and NSAD packet drop comparison (140 nodes)

图 10 DCF 和 NSAD 丢包状况(重负载 140 节点)

5 结束语

本文在数学建模推导基础上获得无线共享信道中负载 l 的最优值所具有的一些重要特性,并利用这些特性设计了一种新型自适应的 NSAD 协议.这种新型协议使网络节点自适应地通过优化 MAC 层竞争初始窗口来调节节点接入概率达到最优,从而使网络吞吐量逼近最大.同时,算法流程通过在数据帧中携带优化后的窗口值给当前网络中所有其他节点,以达到网络节点竞争窗口计算的一致性,保证了公平性.

在之后的仿真实验中,我们比较了 NSAD 与 DCF 的吞吐量、公平性、链路时延、丢包特性以及对上层 TCP 的影响.大量的仿真实验结果表明,在较重负载情况下,这种算法在吞吐量、公平性、丢包、时延和对上层 TCP 窗口机制的保护各方面,其特性较原 DCF 算法都有较大的提高.

References:

[1] IEEE 802.11. Wireless LAN medium access control (MAC) and physical (PHY) layer specifications, 1999.

[2] Peng Y, Wu HT, Long KP, Cheng SD. Simulation analysis of TCP performance on IEEE 802.11 Wireless LAN. In: Proc. of the Int'l Conf. on Info-Tech and Info-Net. 2001. 520~525.

[3] Ozugur T, Naghshineh M, Kermani P, Copeland JA. Fair media access for wireless LANs. In: Proc. of the IEEE Global Telecommunications Conf. GLOBECOM. Rio de Janeiro: IEEE, 1999. 570~579.

[4] Bianchi G, Fratta L, Oliveri M. Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LANs. In: Proc. of the PIMRC 1996. Taipei, 1996. 392~396.

[5] Cali F, Conti M, Gregori E. IEEE 802.11 protocol: Design and performance evaluation of an adaptive backoff mechanism. IEEE JSAC, 2000,18(9).

[6] Aad I, Castelluccia C. Differentiation mechanisms for IEEE 802.11. In: Proc. of the 20th Annual Joint Conf. of the IEEE Computer and Communications Societies. 2001. 209~218.

[7] Deng DJ, Chang RS. A priority scheme for IEEE 802.11 DCF access method. IEICE Trans. on Communication, 1999,E82-B(1).

[8] Wu HT, Peng Y, Long KP, Cheng SD. A simple model of IEEE802.11 wireless LAN. In: Proc. of the Int'l Conf. on Info-Tech and Info-Net. 2001. 515~520.

[9] Bianchi G. Performance analysis of the IEEE802.11 distributed coordination function. IEEE JSAC, 2000,18(3).

[10] Gallager RG. A perspective on multi-access channels. IEEE Trans. on Inform. Theory, 1985,31:124~142.

[11] Jain R. The Art of Computer System Performance Analysis. John Wiley & Sons, 1991.

附录:定理1的证明.

类似于文献[9],我们用离散时间马尔可夫链对 NSAD 协议建模,如图 11 所示.在这里,我们与文献[9]的不同在于,我们的初始窗口值是通过第 2.2 节中的算法进行调节的.这里,n 是个变量.我们申明采用以下表示法:

$$P\{i_1,k_1|i_0,k_0\}=p\{s(t+1)=i_1,b(t+1)=k_1|s(t)=i_0,b(t)=k_0\},$$

$s(t)$代表退避次数,$b(t)$代表退避时槽数.存在以下概率转移方程:

$$\left.\begin{array}{lll} P\{i,k|i,k+1\}=1, & k\in[0,W_i-2], & i\in[0,n'] \\ P\{0,k|i,0\}=(1-p)/W_0, & k\in[0,W_0-2], & i\in[0,n'] \\ P\{i,k|i-1,0\}=p/W_i, & k\in[0,W_i-1], & i\in[0,n] \\ P\{0,k|n',0\}=1/W_n, & k\in[0,W_0-1], & \end{array}\right\} \quad (16)$$

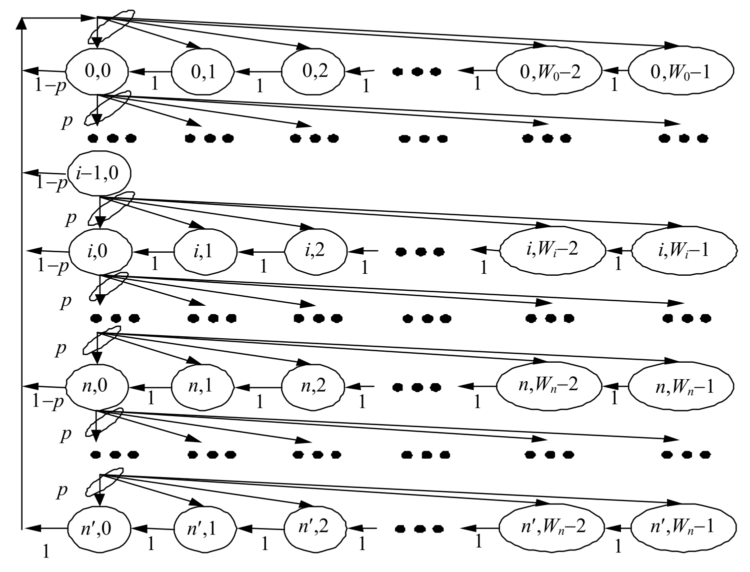


Fig.11 Markov chain model of NSAD window mechanism

图 11 NSAD 窗口机制马尔可夫链建模模型

方程组(16)描述了当前网络中一个活动节点的窗口退避机制.令 $b_{i,k}=\lim_{x\to\infty}P\{s(t)=i,b(t)=k\}$,$i\in[0,n']$,$k\in[0,W_0-1]$为此马尔可夫链稳态分布概率,于是有以下方程组成立:

$$b_{i,0}=b_{i-1,0}\cdot p\to b_{i,0}=p^i b_{0,0}\quad 0<i\le n',$$

$$b_{0,0}=(1-p)\sum_{j=0}^{n'-1}b_{j,0}+b_{n',0},$$

$$b_{i,k}=\frac{W_i-k}{W_i}b_{i,0}=\frac{W_i-k}{W_i}\cdot\begin{cases}(1-p)\sum_{j=0}^{n'-1}b_{j,0}+b_{n',0} & i=0\\ p\cdot b_{i-1,0} & 0<i\le n'\end{cases},$$

$$W_i=\begin{cases}2^i\cdot W_0, & 0\le i<n\\ W_{\max}, & n\le i\le n'\end{cases}\tag{17}$$

通过上述方程组,我们可以将所有 $b_{i,k}$ 表示为 p 和 $b_{0,0}$ 的函数.由于稳态分布具有归一化条件,于是我们得到:

$$1=\sum_{i=0}^{n'}\sum_{k=0}^{W_i-1}b_{i,k}=\sum_{i=0}^{n'}b_{i,0}\frac{W_i+1}{2}\tag{18}$$

最终我们可以得到:

$$b_{0,0}=\begin{cases}\dfrac{2}{\sum_{i=0}^{n'}p^i+W_0\sum_{i=0}^{n'}(2p)^i}, & n'\le n\\[2ex] \dfrac{2}{\sum_{i=0}^{n'}p^i+W_0\sum_{i=0}^{n-1}(2p)^i+W_n\sum_{i=n}^{n'}p^i}, & n'>n\end{cases}\tag{19}$$

值得说明的是,事实上,在 NSAD 协议中,最大重传次数大于窗口增长次数,即 $n'>n$.由于当退避计数器计数到 0 时,站点试图向空闲信道发包,故站点试图接入信道的概率等于退避时槽记数值为 0 的所有状态稳态概率之和:

$$\tau=\sum_{i=0}^{n'}b_{i,0}=b_{0,0}\cdot\sum_{i=0}^{n'}p^i=\begin{cases}\dfrac{2\cdot\sum_{i=0}^{n'}p^i}{\sum_{i=0}^{n'}p^i+W_0\sum_{i=0}^{n'}(2p)^i}, & n'\le n\\[2ex] \dfrac{2\cdot\sum_{i=0}^{n'}p^i}{\sum_{i=0}^{n'}p^i+W_0\sum_{i=0}^{n-1}(2p)^i+W_n\sum_{i=n}^{n'}p^i}, & n'>n\end{cases}\tag{20}$$

联合第 2.2 节中的式(11)可知:

$$\tau_{\rm opt}=\frac{\sqrt{[N+2(N-1)(T_c^*-1)]/N}-1}{(N-1)(T_c^*-1)}\approx\frac{1}{N\sqrt{T_c^*/2}}.$$

代入式(20)中,取 $n'>n$,稍加整理,并注意到 W_0 即 $W_{\rm init}+1$,W_n 即 $W_{\max}+1$,易得定理 1 结论. □